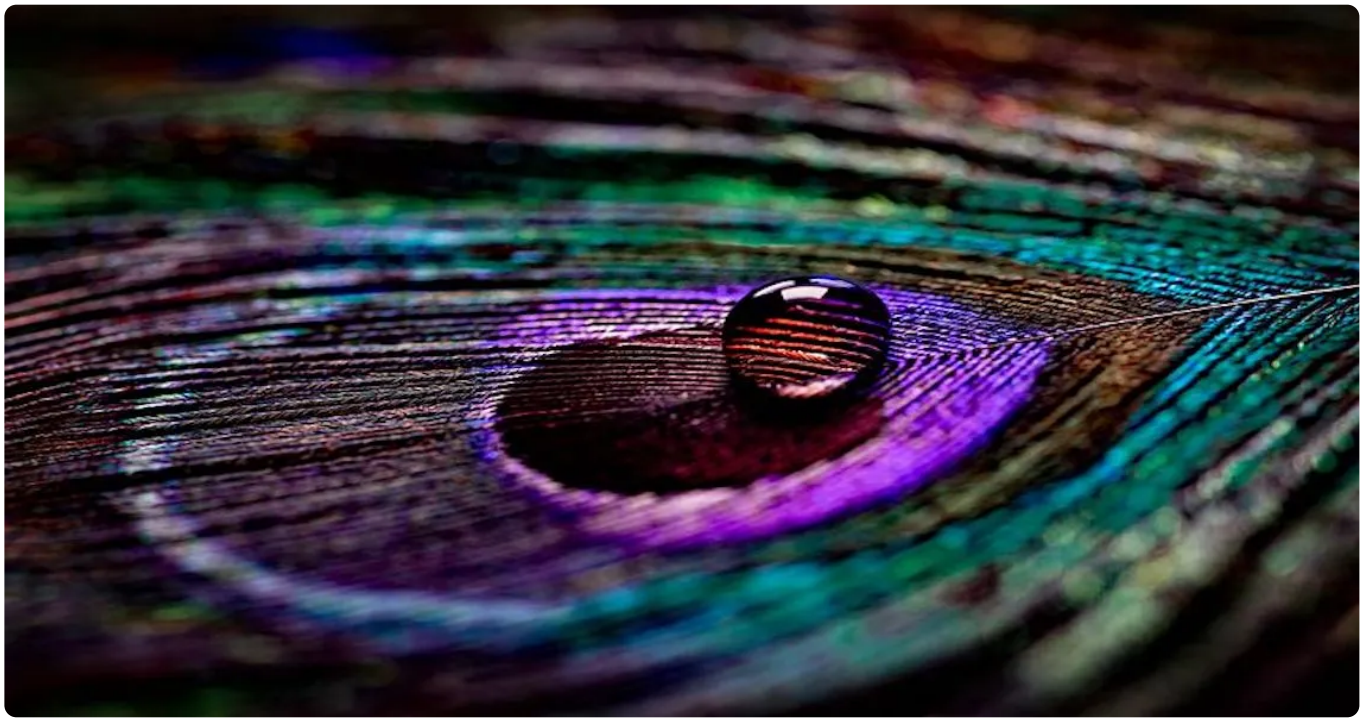
开篇词 | 全面拥抱 Hooks，掌握最新 React 开发方式

2021-05-24 王沛

《React Hooks 核心原理与实战》 课程介绍 >



讲述：王沛
时长 12:40 大小 11.61M

你好，我是王沛。欢迎和我一起学习 React。

我在前端领域深耕 15 年了，先后工作于 IBM 和 SAP，目前在 eBay 担任中国区资深研发技术专家的职位。这期间，既做过前端框架的自研，也构建过大型前端开发的项目。

其实，早在上学阶段，我就对前端开发很感兴趣，研二时就出版了当时国内第一本专门讲 Web 2.0 的书籍：《征服 Web 2.0: Ajax 开发技术详解》。之后的工作也一直专注于前端框架的开发。可以说，我经历了前端的整个发展历史，也深知前端开发的各种痛点、难点，并见证了它们是如何被一个个出现的框架和工具不断攻破的。

而在前端不断推陈出新的技术中，React 的出现让我眼前一亮，它解决了很多前端开发的本质性问题，如数据流、状态维护等，于是我也开始将 React 应用在自己的项目中。经过

数十个大型项目的实操，我把自己对 React 的一些可重用的经验，做成了开源项目，比如 Rekit、React Form Builder 等，在 GitHub 上有超过 5000 个 Star。

同时，作为国内最早一批的 React 实践者，我也不断将自己使用 React 的经验，通过 InfoQ 专栏、QCon 演讲和诸多同行分享，推动了 React 在国内的普及；每年也都会在公司内部开设培训课程，帮助各个团队迅速掌握 React 的常见设计模式和最佳实践。

随着对 React 的深入研究、实战操作、案例积累，以及输出分享，我拥有了一套从核心原理到实战的 React Hooks 知识总结。所以这一次，我想把我多年积累的经验分享给你。

React，前端开发的正确打开方式

和其他技术一样，前端开发技术在不断地更迭变幻。正因如此，面对某个新技术时，我一直抱着非常谨慎的态度。

但是 React 的出现，让我感觉这才是前端开发的正确打开方式。这主要是因为**在当时它有两个非常颠覆式的创新点**。

第一个是**推出了虚拟 DOM 机制**，大幅度降低了 UI 上的操作难度。

React 会自动用最优的方式将虚拟 DOM 更新到真实的 DOM 节点，这样一来，我们就不需要再使用浏览器 API 来维护复杂的 DOM 结构了，这就简化了我们在 UI 上的操作。

第二个是 **JSX 语法的使用**，不仅使代码前所未有的简洁和直观，同时也减轻了内部沟通成本，提高了维护效率。

React 用类似 HTML 标记的方式去实现组件化，数据变化以后，UI 也会自动变化，我们再也不用去关注实现细节了，而且保证了 UI 的一致性。

这两个机制完美地解决了当时前端开发的痛点：模板语言不够灵活，DOM API 又太繁琐，且难以维护。因此， React 在 2013 年首次宣布以后，就迅速获得了大家的认可，成为了最为流行的前端框架。

从下面这张图也可以看到，React 自诞生以来，下载量就处于一直上升的趋势。这在前端轮子倍出的背景之下，可以说是一个不俗的成绩了。

值得注意的是，在其他前端框架发生翻天覆地变化的同时，React 的 API 则非常稳定。所以在我看来，无论你是**专职从事前端开发，还是以后端为主，但也需要自己做一些 UI 界面，React 作为目前最为流行的前端框架，基本上都是你的首要选择。**

但即便在这样稳定的背景下，React 还是在 2019 年 2 月发布的 16.8 版本中引入了 Hooks 概念。在让人耳目一新的同时，也让广大早就恐惧于各种新轮子的前端同学心头一紧，产生本能的抗拒：

> Class 组件不是挺好的吗，为什么又要学一个新轮子？
>
> Hooks 刚出来一两年，会不会昙花一现？
>
> 现在项目代码都是 Class 的，难道都得改一遍吗？

我先给你吃一颗定心丸，这种担忧完全没有必要，这不仅不会增加你的使用难度，反而还**会拓展你的 React 开发思路，为你提供了一种更多的选择。**

全面拥抱 React Hooks，让前端代码结构更加清晰

事实上，在 Hooks 刚出来时，我带领的团队中也有很多同学对 Hooks 产生了抗拒心理。这很正常，毕竟，我们初次使用任何新技术时大都会有怀疑心理，或者是畏惧。正如你第一次走一条不熟悉的路时，总会觉得前路漫漫，有点儿迷茫。

即便如此，出于对 React 官方团队的信赖，我们还是决定开始尝试。惊喜之处就在于，在我们不断了解和使用之后，逐渐发现 Hooks 真是太好用了。原来很复杂的一些问题，现在用 Hooks 就可以很优雅地解决！

举个例子来说。在 Hooks 出现之前，**业务逻辑的重用**可以说是 React 开发的一大痛点和难点。比如，你需要在组件中去监听窗口大小的变化，以便在布局上做调整。那么我们就得在类组件的不同生命周期中做事件监听的绑定和解绑。

其中的困难和复杂可想而知，毕竟，这种需要分散在不同方法中的逻辑是很难去重用的。因此，我们不得不借助高阶组件这样复杂而又晦涩的设计模式来解决。

但现在呢，有了 Hooks，我们就能很轻松地应对了。我们用 Hooks 就可以非常自然地去解决。我们只需要通过 Hooks 的方式进行封装，将其变成一个可绑定的数据源。这样当窗口大小发生变化时，Hook 的组件会自动重新渲染。同时，代码也更加易懂，并且不会产生额外的组件节点。

可以说，引入 Hooks 的概念之后，函数组件就具备了状态管理、生命周期管理等能力，**几乎可以实现原来的 Class 组件具有的所有能力**。函数组件能和已有 Class 组件无缝衔接，一起工作。这样老的代码无需做任何改变，新的代码可以根据具体情况采用 Hooks 的方式去实现。这些都大大加速了 Hooks 的普及。

而且，函数组件结合 Hooks 带来的**简洁的写法，以及直观的逻辑重用能力**，让函数组件成为了 React 中的一等公民。**目前，在整个 React 生态中，几乎所有开发项目都已经是围绕着 Hooks 展开的了**。

所以，在这次课程中，我会**从函数组件和 Hooks 的角度出发，带你重新认识 React**，帮助你在深入理解 Hooks 的基础上，使用 Hooks 的思想去开发前端应用，真正享受 Hooks 带来的诸多好处。

不过，对于初次接触 Hooks 的同学来说，肯定会经常遇到各种问题。比如：

为什么编译又不过了？

我的函数组件怎么又死循环了？

Redux 的 Store 要怎么访问？

有这些问题是很正常的，因为**从 Class 组件到函数组件的转变，不仅仅是一个写法的区别，更是整个开发思路的转变。而我们只有学会用 Hooks 的角度去思考问题，才算把这条路真正走顺。**

那么，在这个课程里面，我会将我在使用 Hooks 的过程中，以及作为面试官在面试的过程中遇到的、看到的各种痛难点问题，以及最需要理解的核心原理呈现出来，帮助你把路走顺，且少走弯路，快速掌握并使用 React Hooks 思想进行前端开发。

想用好 React，你该这么学

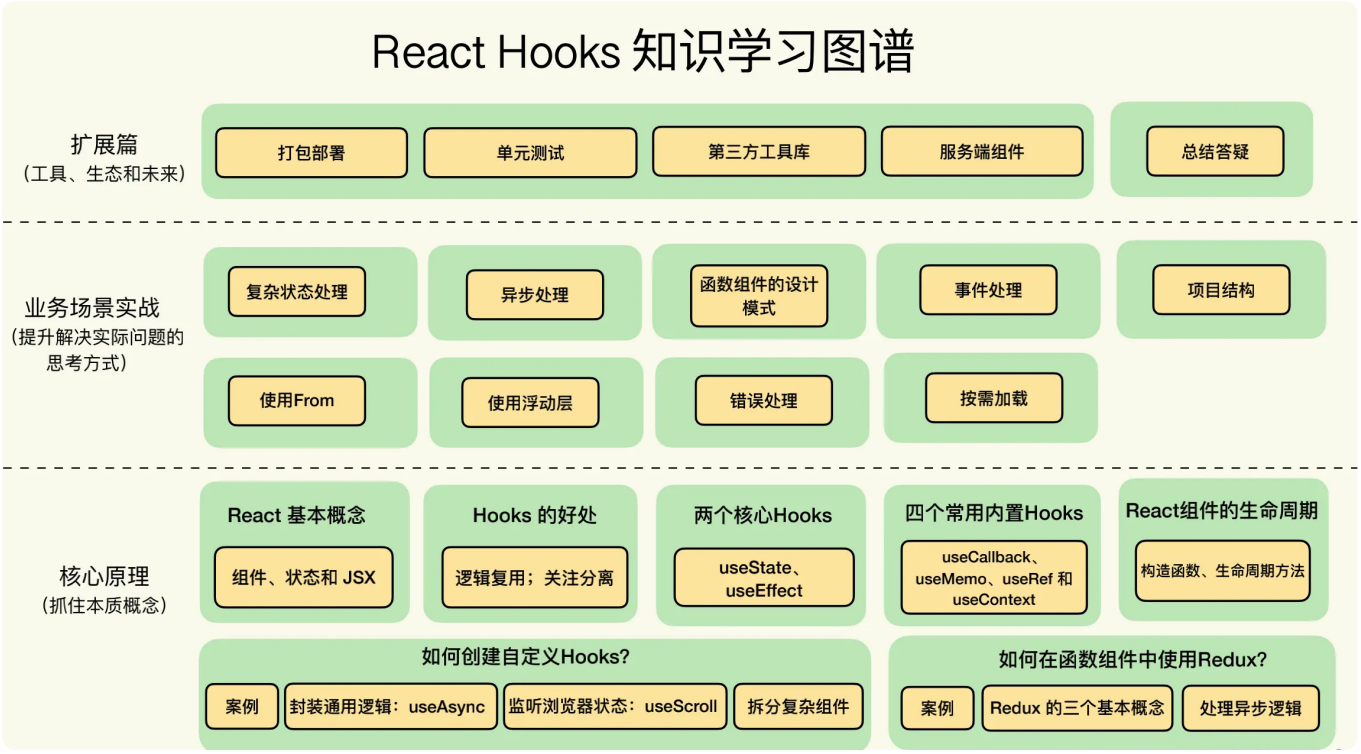
现在，你一定觉得 Hooks 简直太神奇了，迫不及待地想要学习。别急，深入技术点之前，你得知道科学的学习路径。

对于 React Hooks 而言，在我看来最重要的是准确地知道 Hooks 的功能边界，也就是知道它能做什么。否则，就有可能陷入什么功能都想用 Hooks 去做的，或者是在最适合使用 Hooks 的功能中，却想不到用它去处理。

在建立起对 Hook 的全局认知之后，就要掌握**核心概念和原理**了。任何技术，脱离了原理本身，学到的大概率只是皮毛。原理或许不能让你即学即用，但这却是你**应对实际复杂业务问题的关键能力**。

当然，**把原理和实际业务场景结合起来**，才能让自己在 Hooks 进阶这条路上走得更快、更稳。所以，你就必须清楚日常开发中的具体场景，在真实的场景中透彻理解 Hooks 是如何解决问题的。而我们最高阶的目标，就是让你**在实战中掌握 Hooks 的思考方式，拥有举一反三的能力**。

刚刚说的这些，其实也正是咱们这个课程的核心设计理念。据此，我把学习这门课的路径画成了一张图，你可以对照着来学习。



课程是如何设计的？

接下来，我就说说这门课具体是怎么设计的。整个课程会分为三个部分。

基础篇：抓住问题本质，提高解决复杂问题的能力

刚才我们讲到，原理的学习是必要且重要的，那么，为了让你快速消化吸收这些原理，学得有趣，我也做了一些精心的小设计。

在“基础篇”，我会抽丝剥茧，从“问题的本质是什么”这一角度出发来讲解每个机制，包括：React 为什么要发明 Hooks？useEffect 就是生命周期函数吗？等等。理解了最核心的那个问题，你也就拥有了举一反三的能力。

实战篇：业务场景驱动，掌握解决实际问题的思考方式

在理解核心概念的基础之上，我们就拥有了解决问题的抓手，接下来，我会带你进入实战部分。

在这里，我精心设计了 9 个不同的典型业务场景，并基于 React Hooks 去剖析一些常见设计模式。相信通过实战锻炼，你会快速掌握这些通用原则和最佳实践，形成自己的知识库。在遇见类似的场景时，可以灵活调取。

扩展篇：了解工具、生态和未来，开拓你的技术视野

我们学习 Hooks，是为了更好地使用 React，因此，在最后一个模块，我们必须要将视角扩展至 React 的整个生态。

我会给你介绍 React 应用该如何打包，整个生态有哪些可用的工具和资源，该如何测试，以及未来会有哪些发展的趋势。有了完整体系的积累，你才能真正独当一面，完成整个应用的开发、测试、打包和部署。

最后，希望这门从函数组件角度出发的 React Hooks 课程，不仅能够帮助你全面掌握 React 的开发技能，同时也能够给你一个很好的进阶起点，让你在 React 的开发路程上更加高效地前进。

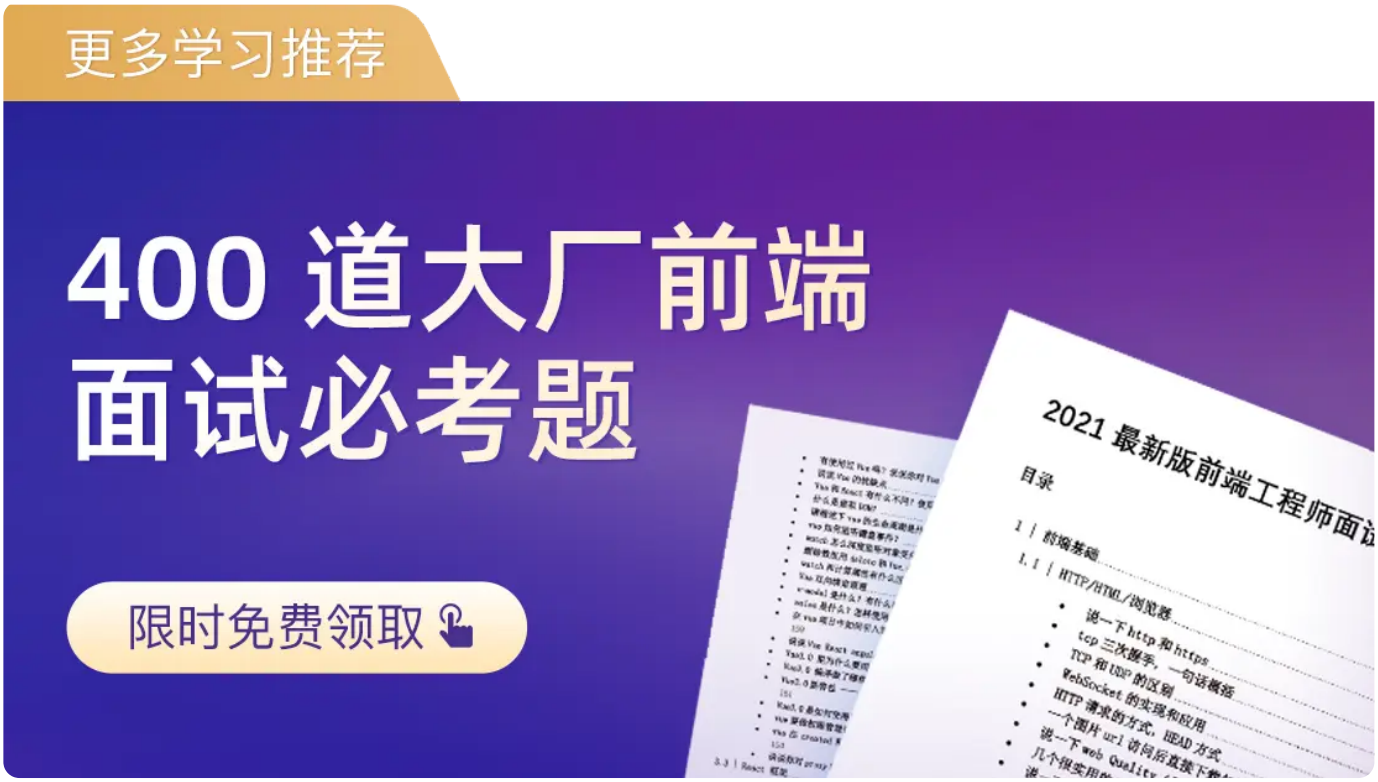
当然，如果你在学习过程中有任何问题或者心得，也欢迎在留言区和我交流互动，我们一定可以共同进步。

分享给需要的人，Ta订阅后你可得 20 元现金奖励

赞 24 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

下一篇 01 | 认识 React：如何创建你的第一个 React 应用？



精选留言 (11)

写留言

Redmi_18117543951 置顶
2021-05-24

春招面试的过程中只要提到React话题，几乎每次都会被问到Hooks的相关内容，所以一看到专栏就买下来了，希望秋招时我能信心满满地跟面试官打几个回合😁

展开

作者回复: 对此我很有信心 ☺

2 5

busy–1992
2021-05-25

课程没视频吗？

展开

作者回复: 这次是文字形式哦。

1 2

琼斯基亚

2021-05-25

他来了他来了，他带着干货走来了！王老师超甜……（期待已久的hooks）

作者回复: 甜。。

2

Selfimpr

2021-05-25

王沛老师的github地址可以分享一下吗？

展开

作者回复: 我个人的 github 是 https://github.com/supnate
这个课程的代码地址是在 codesandbox 上：https://codesandbox.io/s/react-hooks-course-20vzg

3

墨川

2021-06-20

□老师，单学一个react能找到前端工作吗？企业招聘react工程是要求会webpack,有必须要单独学下□打包工具技术吗，如果想成为一个能满足企业实际要求的react工程师还需要哪些技能？

作者回复: 对于 webpack 有一个基本的了解我认为就可以了，因为一旦配置几乎就不太需要太多改动。我理解 webpack 的定位，就类似于，写 java 如何配置java 编译器，写 swift，如何配置 swift 编译器。这个对于业务开发人员来说，不一定需要非常深的了解。

对于企业上要求的 react 开发，最主要还是对各种概念和框架是否能够熟练正确使用。以及如何进行组件划分，状态划分，要思路清晰，语义明确。

2

我爱北京天安门

2021-06-04

看到react果断入了，买完发现是之前45讲作者，很棒

展开

作者回复: 多谢支持！

2

程

2021-05-24

hook用了一年了，来学习一下王老师的高见。刚学react的时候就是看王老师的专栏

作者回复: 感谢支持！既然已经用了一年，后面课程希望也能在留言里多分享交流☺

2

守望

2021-06-04

被同事说我的代码像套娃一样，所以来找老师求解react怎么写

作者回复: 尽量写流程式的代码，会更容易维护~

1

Juntíng

2021-05-26

学习 React 的时候看的王老师的课，之前更新 React Hooks 的时候，老师提到过后续的更新，现在终于等到了，继续和王老师学习。

展开

作者回复: 感谢支持~

1 1

MirGhojam

2021-05-26

王老师回来了，oh yeah，

展开

作者回复: 感谢支持 ☺

刘大夫
2021-05-24

加油

展开

作者回复: 一起加油